Tracking in LDC Detector, including VXD, FTD, SIT & TPC Implementation within MarlinReco package

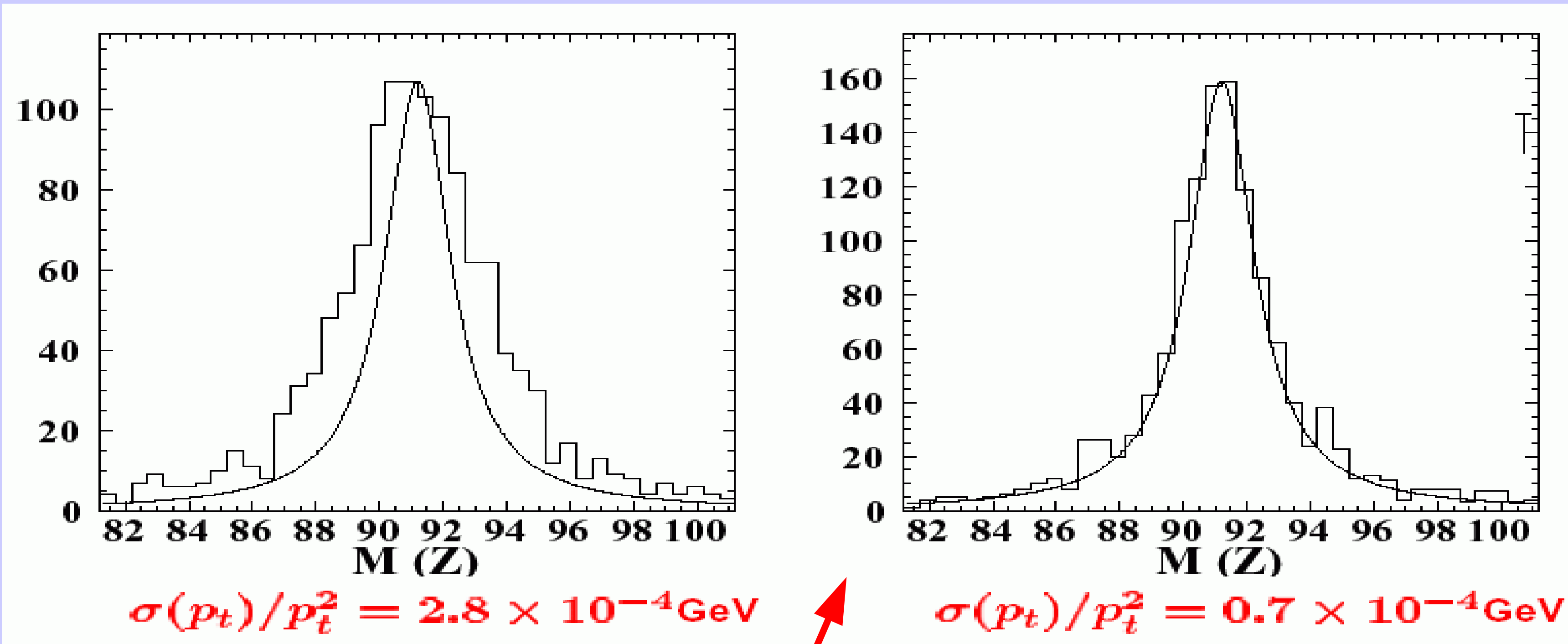
A.Raspereza, A.Frey, X.Chen ***MPI-Munich***

ECFA-ILC Workshop
Valencia November 2006

Outline

- Overview of LDC Tracking System
- Tracking Software Package
  - Digitization
  - Tracking in Silicon Detectors
  - Tracking in TPC
  - Merging TPC & Si tracks ⇒ Combined tracking
- Performance
- Track Fitting
  - Simple Helix Fit
  - Delphi Fit (Kalman fitter accounting for interactions)
- Conclusion & Outlook

LDC Tracking (Performance Goals)

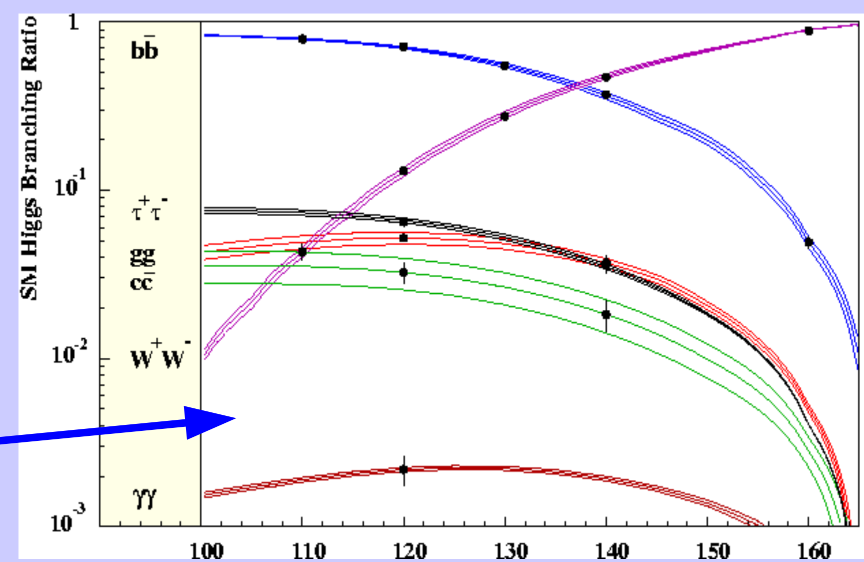


- Benchmark goals
  - $\delta p_T / p_T^{\ 2} \simeq 2 \cdot 10^{-4}$ (TPC only)
  - $\delta p_T / p_T^{\ 2} \simeq 10^{-4}$ (TPC+VXD+SIT)

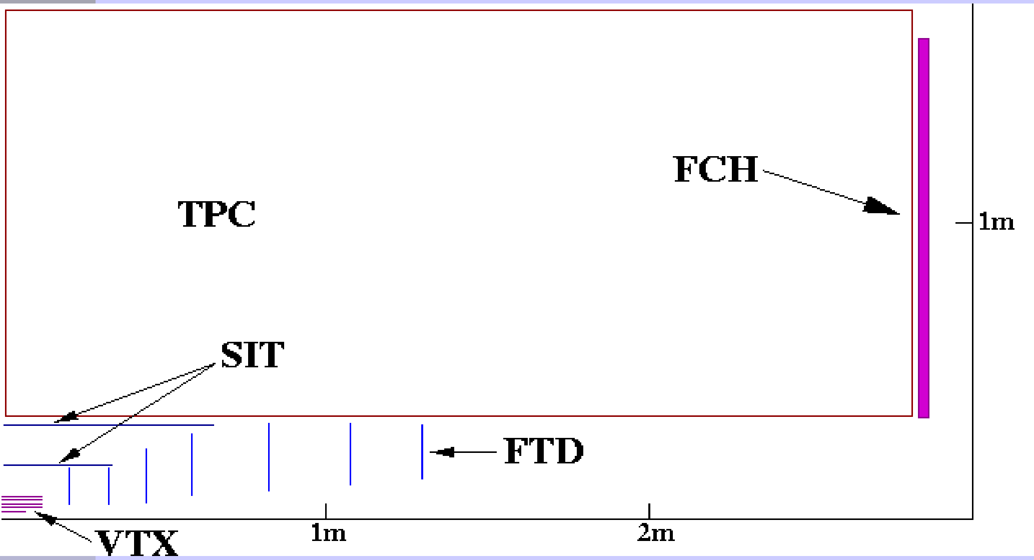
(motivated by $ZH \rightarrow l^+ l^- X$ analysis)

  - $\sigma_{d0} = 5\text{um} \oplus 10\text{um} / p_T[\text{GeV/c}] \cdot \sin^{3/2}\theta$

(to ensure good b- and c- tagging)



Role of Each Subdetector in Light of Tracking



In terms of track parameters
($\Omega,\varphi,\tan\lambda,d_0,z_0$)

- $d_0,z_0 \Rightarrow$ VTX
- $\Omega,\varphi,\tan\lambda \Rightarrow$ Central : VTX+SIT+TPC
- $d_0,z_0,\Omega,\varphi,\tan\lambda \Rightarrow$ Forward : FTD

- TPC : tracking in large gaseous volume; reconstruction of (P_x,P_y,P_z) + dE/dx measurement
- VTX : precise reconstruction of track parameters in vicinity of IP (d_0,Z_0); additional measurement points to improve momentum re-construction
- FTD : tracking in forward region
- SIT : bridge between TPC & VTX (crucial for V^0 finding)
- FCH : facilitates efficient track-calorimeter cluster matching be-yond TPC endplate

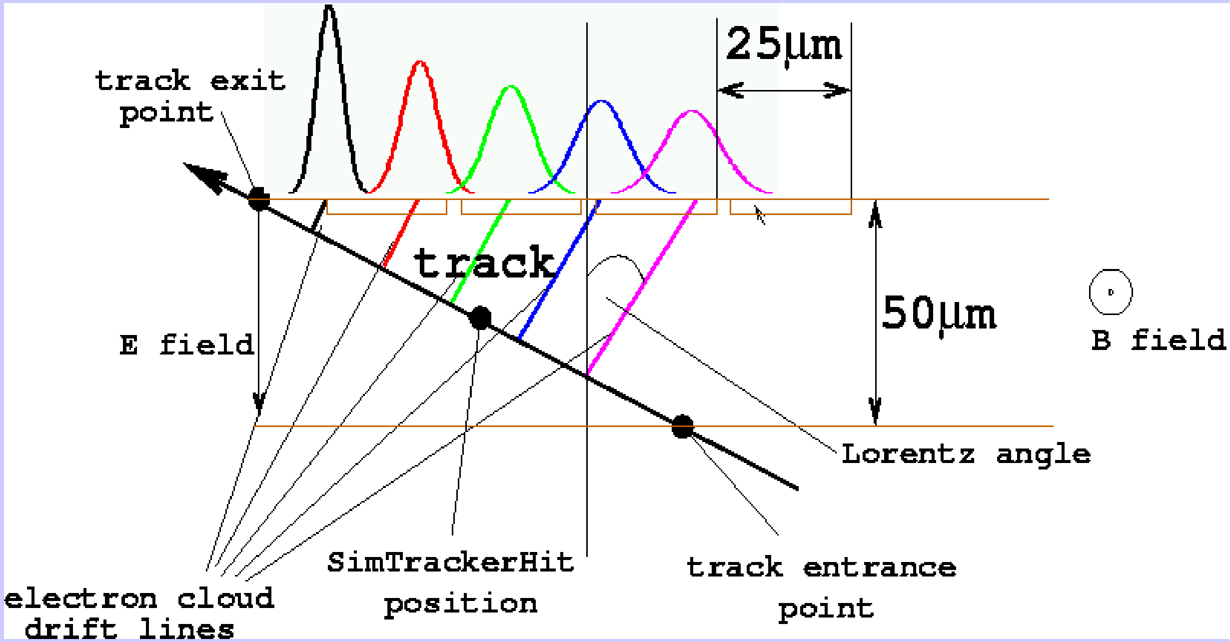
Part of MarlinReco Package Related to Tracking

- Creation of detector geometry & material database needed for patrec and track fitting ⇒ Marlin Processor `MaterialDB`
- Tracker Hit Digitization ⇒ Marlin Processors `VTX/FTD/TPCDigiProcessors; VTXDigitizer`
- Tracking in Silicon Detectors :VTX+SIT+FTD ⇒ Marlin Processor `SiliconTracking`
- Tracking in TPC ⇒ `LEPTrackingProcessor`
- Combining TPC and Si tracks, assigning Si Hits to left-over (unmatched) TPC tracks ⇒ `FullLDCTracking` Processor
- Utility Classes in the Track reconstruction procedures (MarlinUtil package) ⇒ `HelixClass`, `TrackExtended`, `TrackerHitExtended`, `ClusterShapes` classes *etc*

MarlinReco : Digitization of SimTrackerHits

- Two approaches for Tracker Hit digitization
  1) Straightforward gaussian smearing of SimTrackerHit position (TPC/VTX/FTD/DigiProcessor in Marlin) based on specified (apriori-known) spatial point resolution (r-ϕ and Z resolutions)
  2) Detailed digitization based on features of VXD technology and readout

- Example : DEPFET
  - VTXDigitizer ⇨
  - Geometry related info needed for digitization is accessed via GEAR



- Option to emulate/overlay beam induced background hits (integration time is specified for each detector layer as external Processor parameter)

Si Tracking (Strategy, Features)

- Tracking in Si Detectors is implemented via Marlin Processor (SiliconTracking, obsolete name VertexTracking), which does
  - ✔ separate tracking in cyllindrical/laddered detectors (VXD-SIT, two SIT layers are treated as outermost layers of VXD)
  - ✔ separate tracking in planar discs (FTD)
  - ✔ combination of track segments found in VXD-SIT & FTD
  - ✔ track fitting, determination of track parameters @ PCA
- Inputs/Prerequisites
  - ✔ Collection of digitized TrackerHits (VXD/FTD/SIT)
  - ✔ Optionally, constructed RAM-wise material database needed for Delphi Fit (explained later)
- Output
  - ✔ Collection of tracks : track parameters + cov. matrix + list of hits contributing to a given track
- Code is accompanied by doxygen documentation (detailed tex/ps/pdf documentation is foreseen)

Algorithm (General Description)

- In each branch (VXD-SIT or FTD) hit triplets are searched for in different layers (starting from outermost layers), followed by inward search for additional hits in other layers
- Track fit χ^2 as the main criterion to accept track candidate in the branch (tracks are allowed to share only one hit if this is innermost hit in both tracks – handle of possible conversion in the Si layer)
- Track segments in two branches are merged on the basis of combined fit χ^2 criterion ➡ single track containing hits in VXD/SIT & FTD
- Track quality criteria (to suppress fake track rate due to beam induced background)
  - ✗ Total number of hits > 3
  - ✗ At least one hit in SIT if track is found to cross both layers of SIT
  - ✗ At least $N_{FTD}/2$ hits in FTD if track is found to cross N_{FTD} layers of Forward Tracking Discs
  - ✗ Mild cuts on IP $D_0 (Z_0) < 10(20)$ cm (vanishing effect on secondary /tertiary vertex finding in decays of D- and B-mesons)

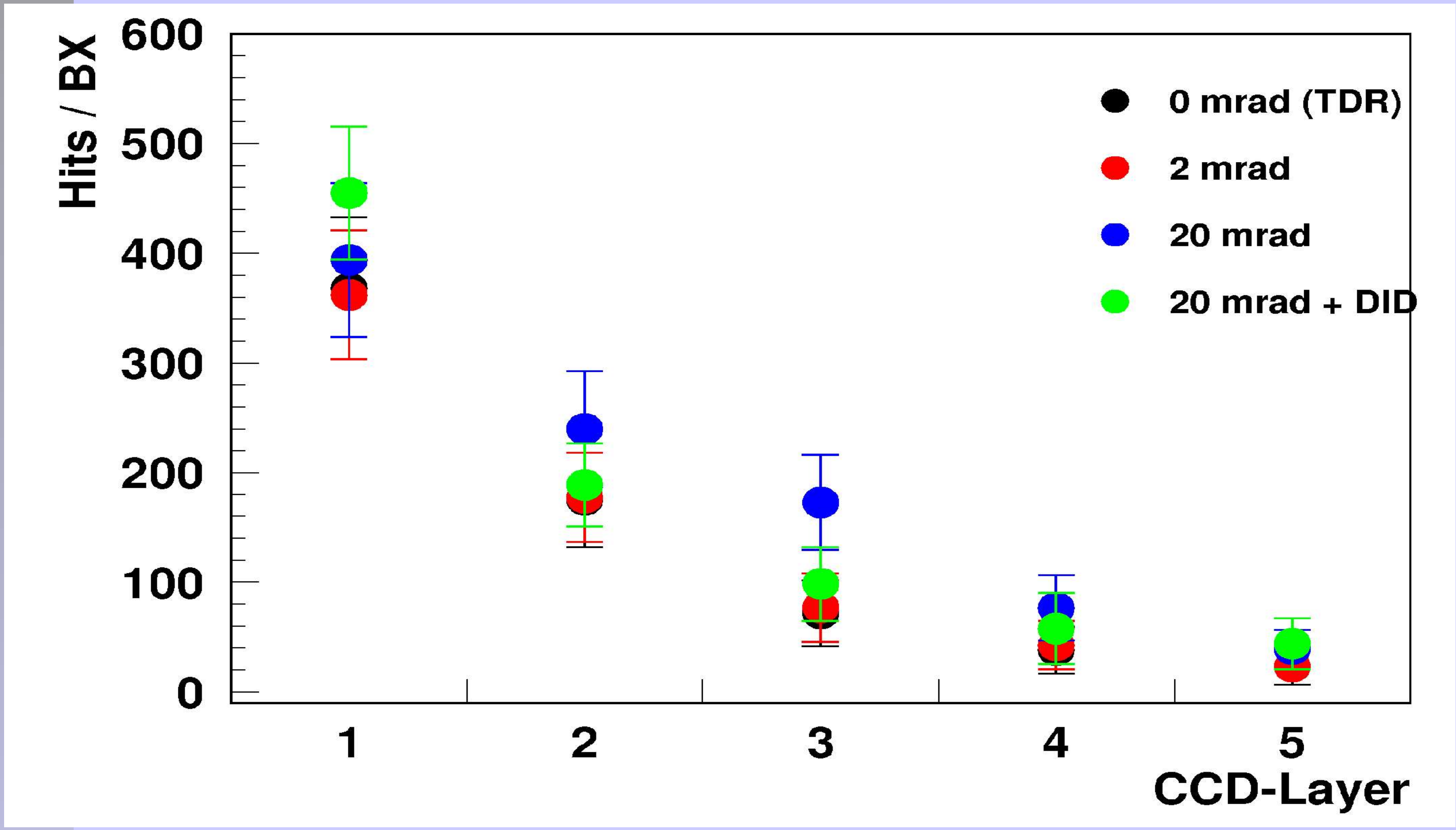
Structure of Code, Dependencies

- Patrec + Track fitting (Marlin processor `SiliconTracking`)
  - HelixClass (MarlinUtil package)
  - tfithl.F (simple helix fit from Brahms tracking package)
  - trkfit.F (interface to Delphi Kalman Fitter)
- An user has an option to chose between simple helix fit and Delphi Kalman Fitter, if second option is chosen an user should include processor “MaterialDB”
  - Reads GEAR and calculates detector material shapes
  - Fills FORTRAN common blocks (on-flight RAM-wise database) storing material shapes, properties
  - Database is accessed by Delphi fitting routines

Performance

- Performance is evaluated on the sample of tt ⇒ 6-jet events in terms of
  - Track finding efficiency
  - Fake track rate
- Distributions are binned in $\log_{10}(P_T)$ & $\cos\theta$
- Study is done for no-background scenario and for scenario assuming integration of 75 BX in the innermost layer of VXD and 150 BX in other layers (as suggested by M.Winter), FTD and SIT are assumed to be capable of tagging every single BX

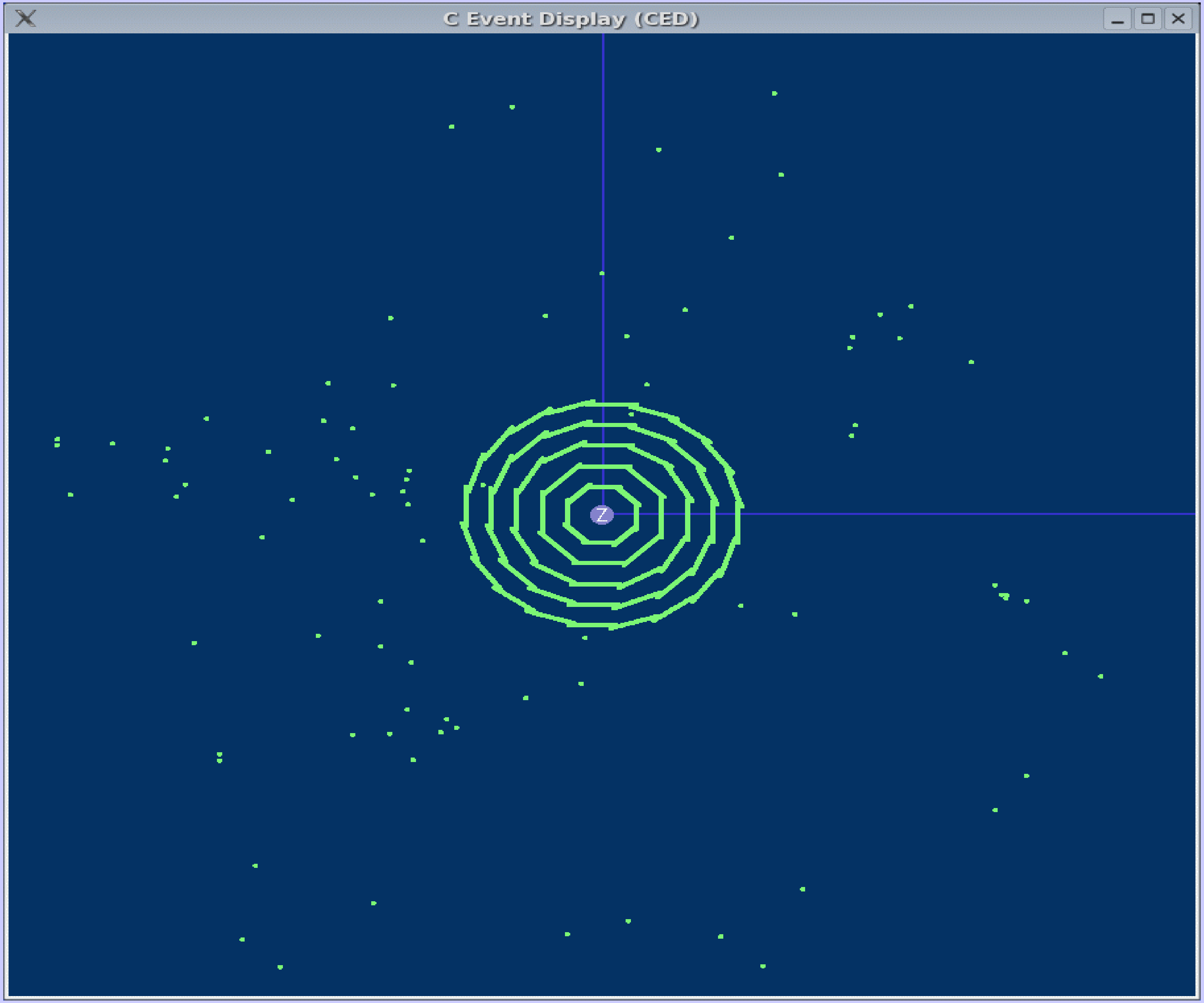
Background Rates



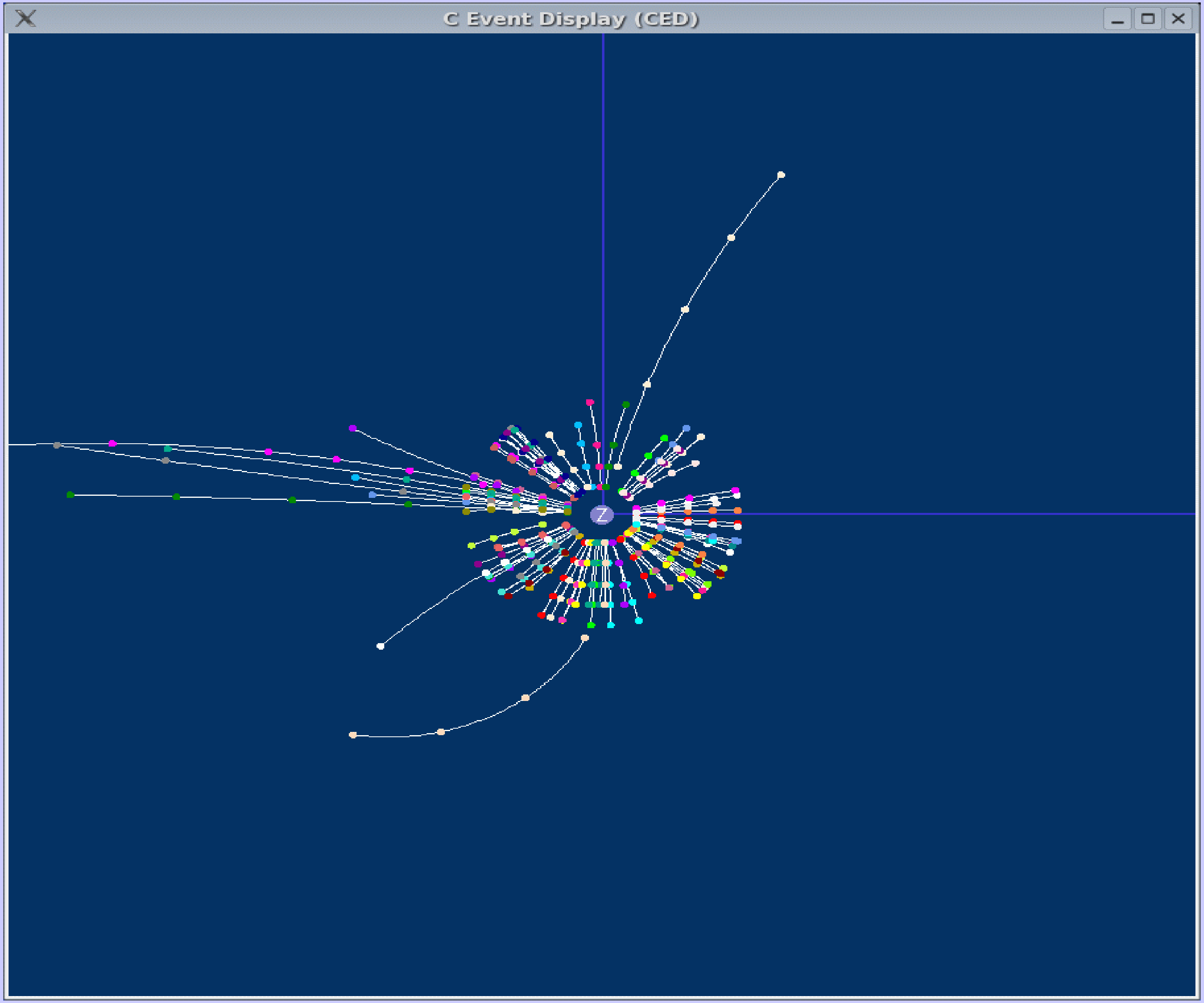
K. Buesser

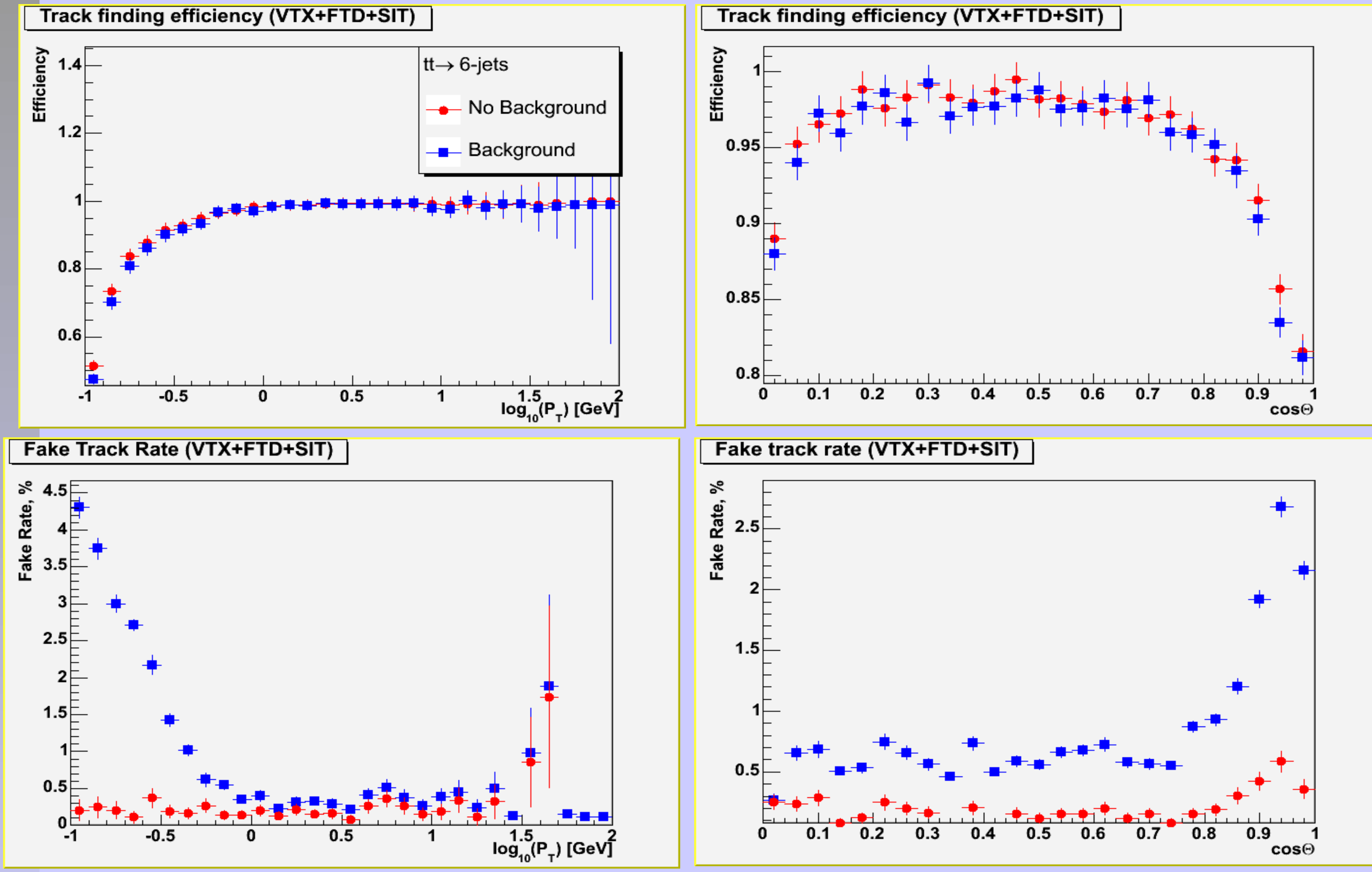
- Assumed rates (0,2 mrad crossing angle)
  - ✔ Layer # 1 : 75 BX ⇒ 28k hits
  - ✔ Layer # 2 : 150 BX ⇒ 28k hits
  - ✔ Layer # 3 : 150 BX ⇒ 13k hits
  - ✔ Layer # 4 : 150 BX ⇒ 8k hits
  - ✔ Layer # 5 : 150 BX ⇒ 6k hits

Event Display with Background Hits ($t\bar{t} \Rightarrow$ 6-jets)



Event Display after Reconstruction (tt⇒6-jets)

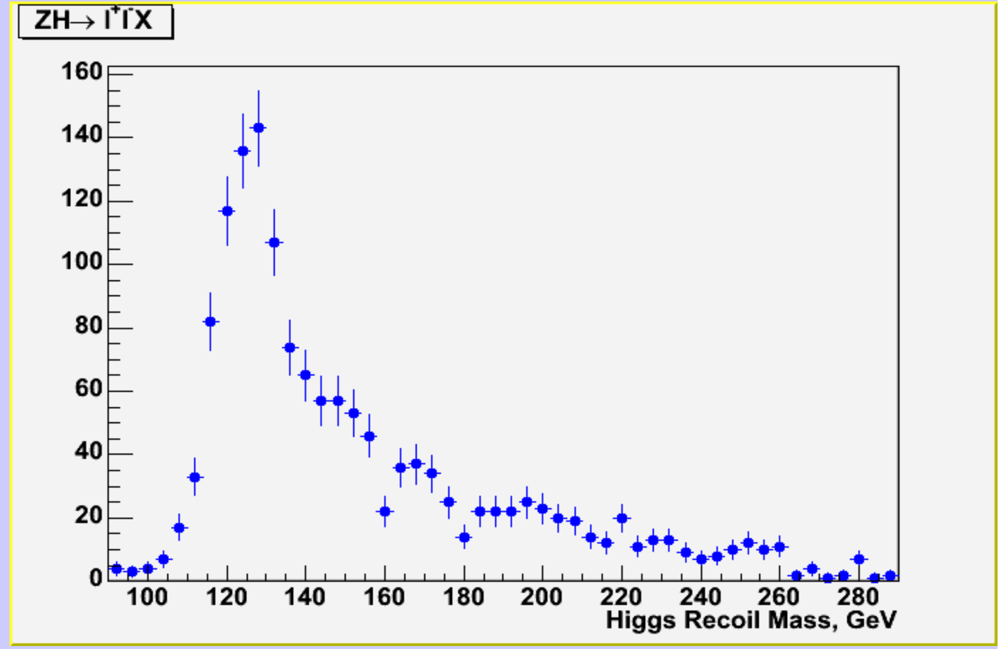


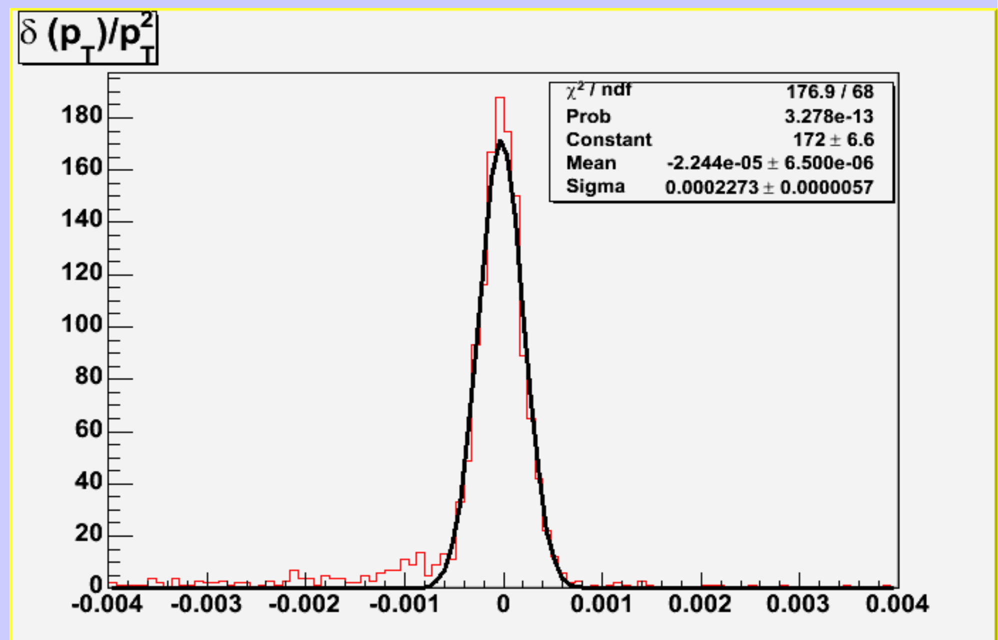


Efficiencies are determined only for those tracks, which leave at least three hits in Inner Tracker (VTX+SIT+FTD)!!!

Tracking in TPC

- Is based on C++ wrappers of LEP codes (author : Steve Aplin)
- Input Collection : TPC Tracker Hits
- Output Collection : TPC Tracks (no co-variant matricies at the moment)
- Track fit with DELPHI fitting routine (multiple scattering and energy loss effects are taken into account)
- Achieved momentum resolution $\delta p_T / p_T^2 = 2.3 \cdot 10^{-4}$
- Mokka model LDC01, TPC Geometry :
  - R_{in} = 37 cm, R_{out}=152 cm
  - Number of pad rows = 184
  - Point resolutions
    - Z : 1mm
    - R-ϕ : [0.16 – 0.01*z/DriftLenght] mm





Full LDC Tracking

- Two steps
1) Matching of Si & TPC Tracks
2) Attachment of non-assigned Si Hits to left-over TPC tracks
- Inputs :
  - Collections of TPC, VTX, SIT & FTD Tracker Hits
  - Collections of TPC & Si tracks
- Outputs :
  - Collection of full LDC tracks (track parameters + covariance matricies), each track contains a list of TrackerHits contributing to a given track
  - Collection of relations between full LDC tracks and MCParticles; each re-constructed LDC track has relations to MCParticles whose hits are in-cluded in track. Each relation is assigned a weigth $\mathbf{w_i = N_i / N}$, where $\mathbf{N_i}$ is the number of hits of i^{th} MCParticle included in a given track and **N** is the total number of hits in a given track
- In the current version track is fitted using simple helix model (implementa-tion of a more sophisticated fitting taking into account energy loss & multiple scattering is in progress)

Full LDC Tracking. Step1
Merging TPC & Si Track Segments

- For each pair of Si and TPC tracks close in momentum phase space $\{ \measuredangle(P_{si}, P_{TPC}) < 5\text{mrad}, |(\Omega_{si} - \Omega_{TPC}) / \Omega_{TPC}| < 5\% \}$ the combined fit of TrackerHits is performed and χ^2 is calculated. Combined track candidate is accepted if $\chi^2/\text{ndf} < 20$
- Combined track candidates are sorted in acsending order of their χ^2
- First combined track candidate in the sorted array is accepted and Si and TPC tracks are marked as used
- The i^{th} combined track candidate in the sorted array is accepted if it containes still unused TPC and Si track segments, otherwise it is discarded. For accepted combined track, Si and TPC track segments are marked as used
- Procedure is repeated until all tracks are accepted or discarded
- Leftover separate TPC & Si track segments are also retained in the final list of full LDC tracks

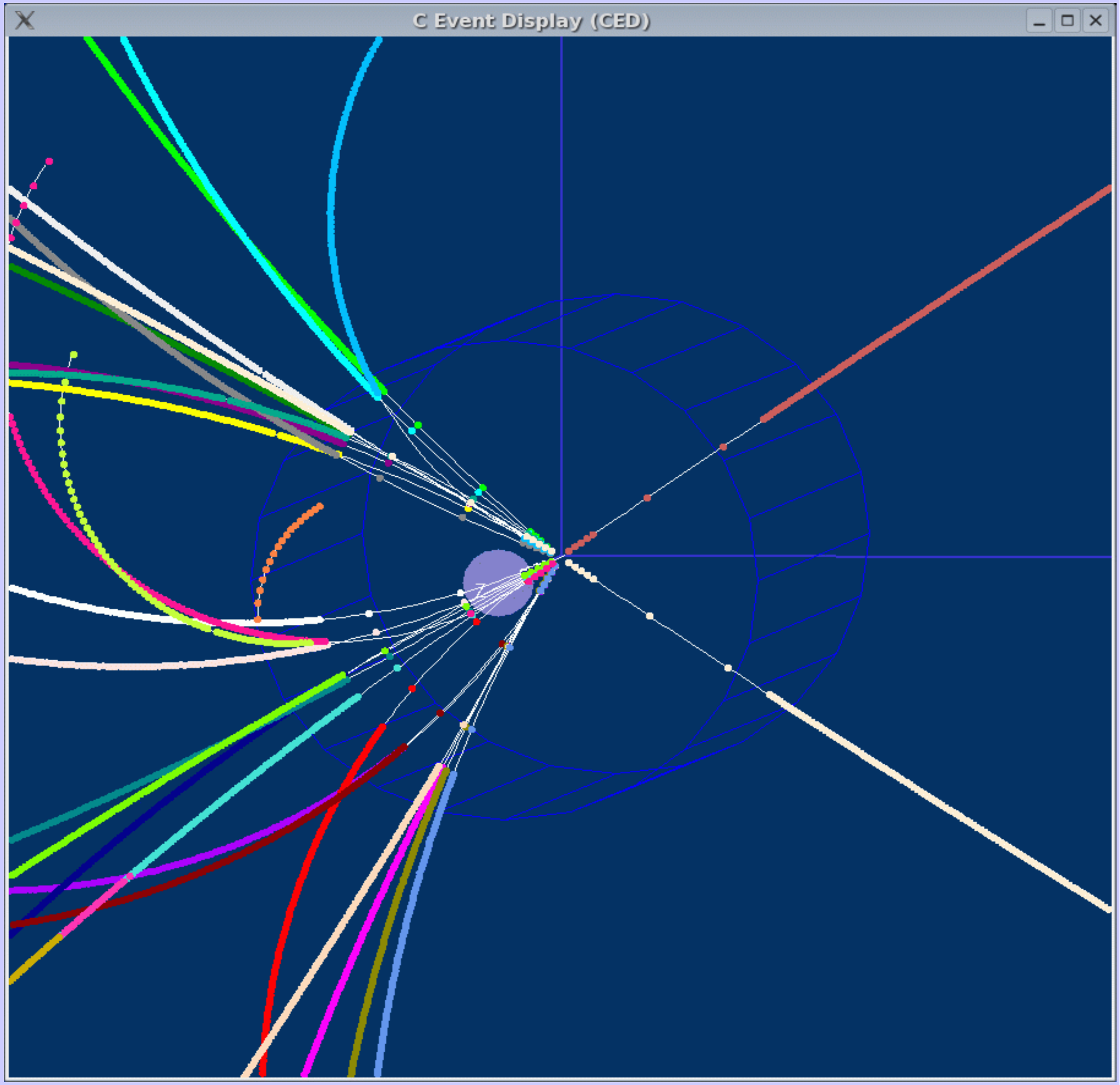
Full LDC Tracking. Step2. Attaching non-assigned Si Hits to Leftover TPC Tracks

- Leftover TPC tracks may contain Si Tracker hits non-assigned to Si track segments if
  - True track contains < 3 TrackerHits in Silicon detectors (for instance, tracks from K^0 decay beyond last layer of VTX but before 1st layer of SIT)
  - SiliconTracking processor failed to reconstruct track segment in Silicon detectors
- Therefore an attempt is made to search for missing Si Hits which may belong to TPC track segments. The search is performed inward starting from the outermost SIT/FTD layer.
- All Si TrackerHits in a given SIT/VTX/FTD layer are tested for their compatibility with leftover TPC track segments. If track refit with Si TrackerHit included gives reasonable χ^2 (χ^2/ndf < 20), then hit is assigned to Track, If more than one hit in a layer can be assigned to the same Track or more than one track pretends for the same hit, we choose the combination which provides better χ^2

Full LDC Tracking

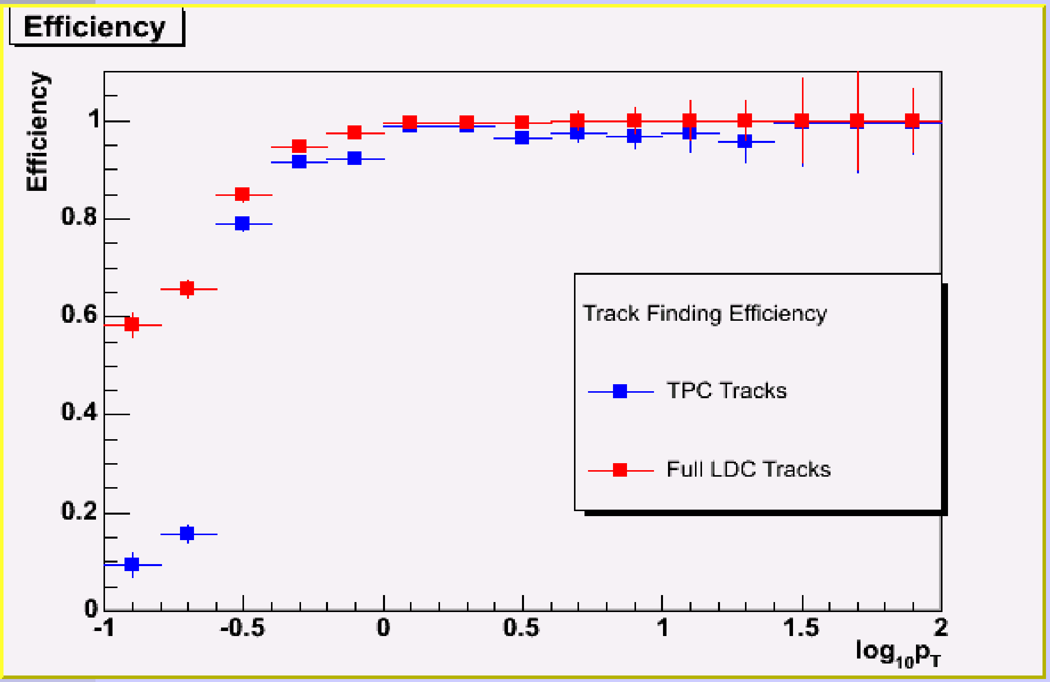
- Only in 0.2% of cases code fails to merge TPC and Si track segments
- Fake track rates is below 0.5%
- Splitted track rate is 2%
  - Mainly loopers in TPC
  - Also tracks, experiencing scattering in Si layers ⇨ sizable change of track parameters at the point of scattering (educative guess: these effects can be eliminated taking into account effects of energy loss and multiple scattering in the fitting procedure)

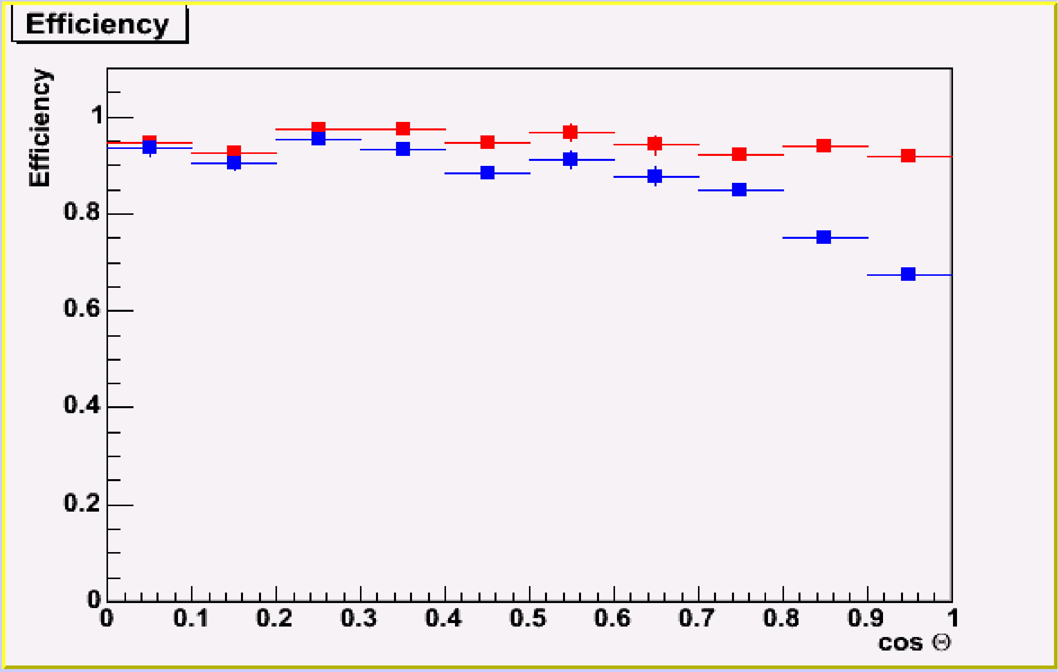
ZH→ $\mu^-\mu^+$bb



Full LDC Tracking Performance

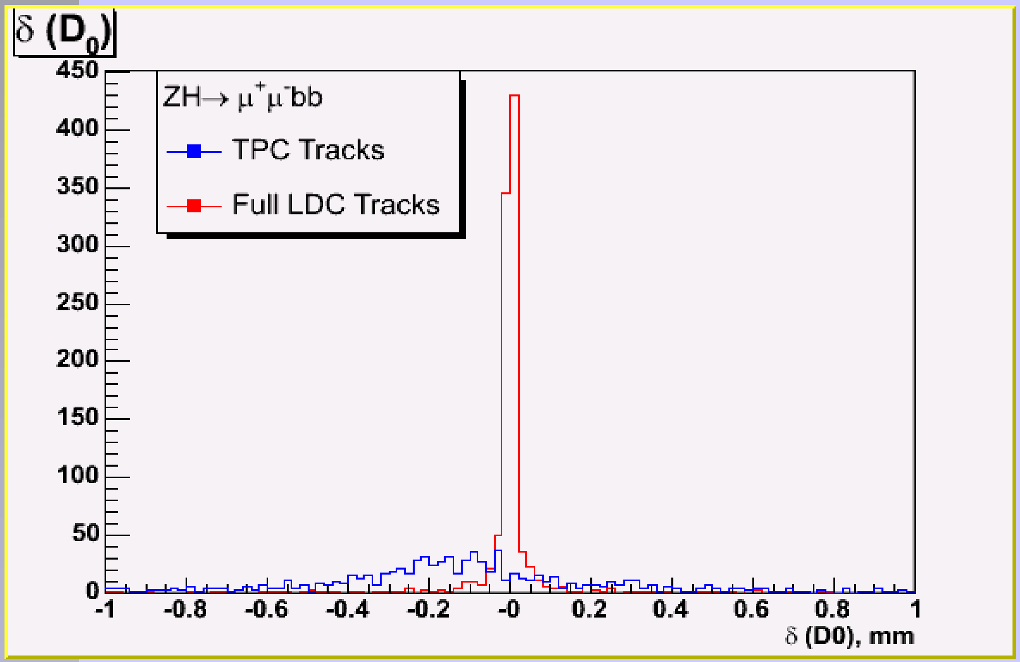
- Track finding efficiency is evaluated as a function of $\log_{10}p_T$ and $\cos\theta$ with the sample of $ZH\rightarrow\mu^+\mu^- bb$ events @ 350 GeV

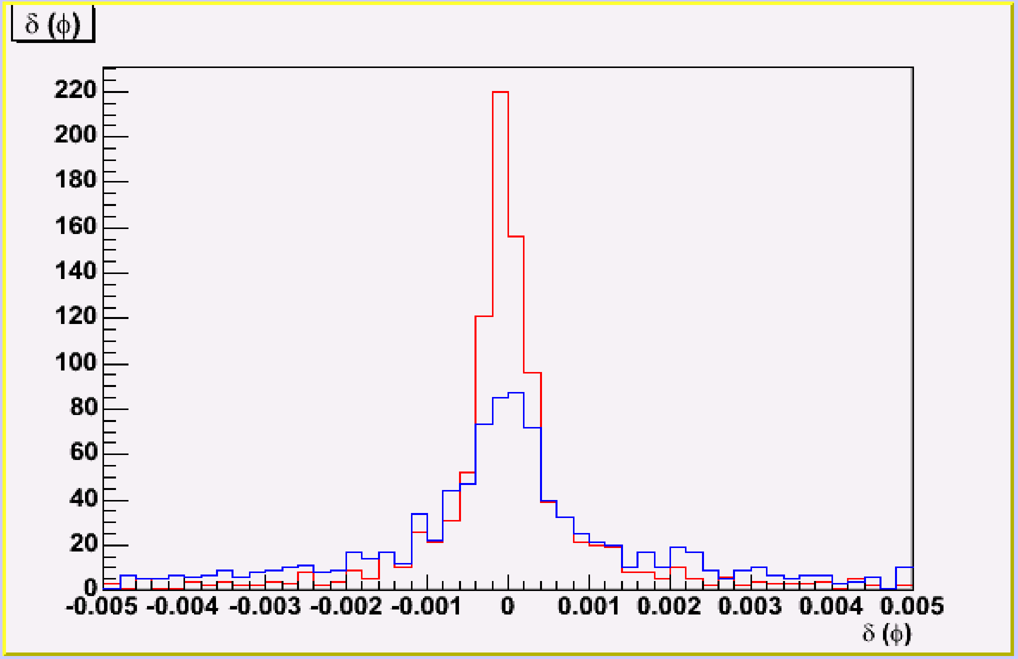


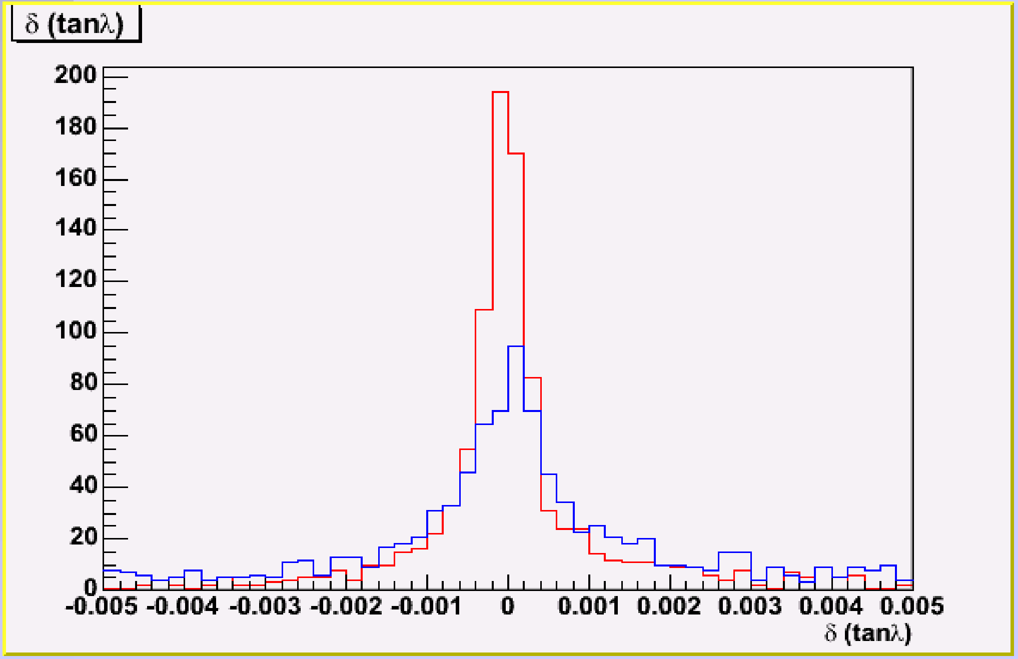


Considerable improvement of efficiency in particular for low p_T and forward tracks!

Full LDC Tracking Performance







Essential improvement in reconstruction of impact parameter (D_0) & direction of particle momenta (parameters φ and $\tan\lambda$)

still with simple helix fit!

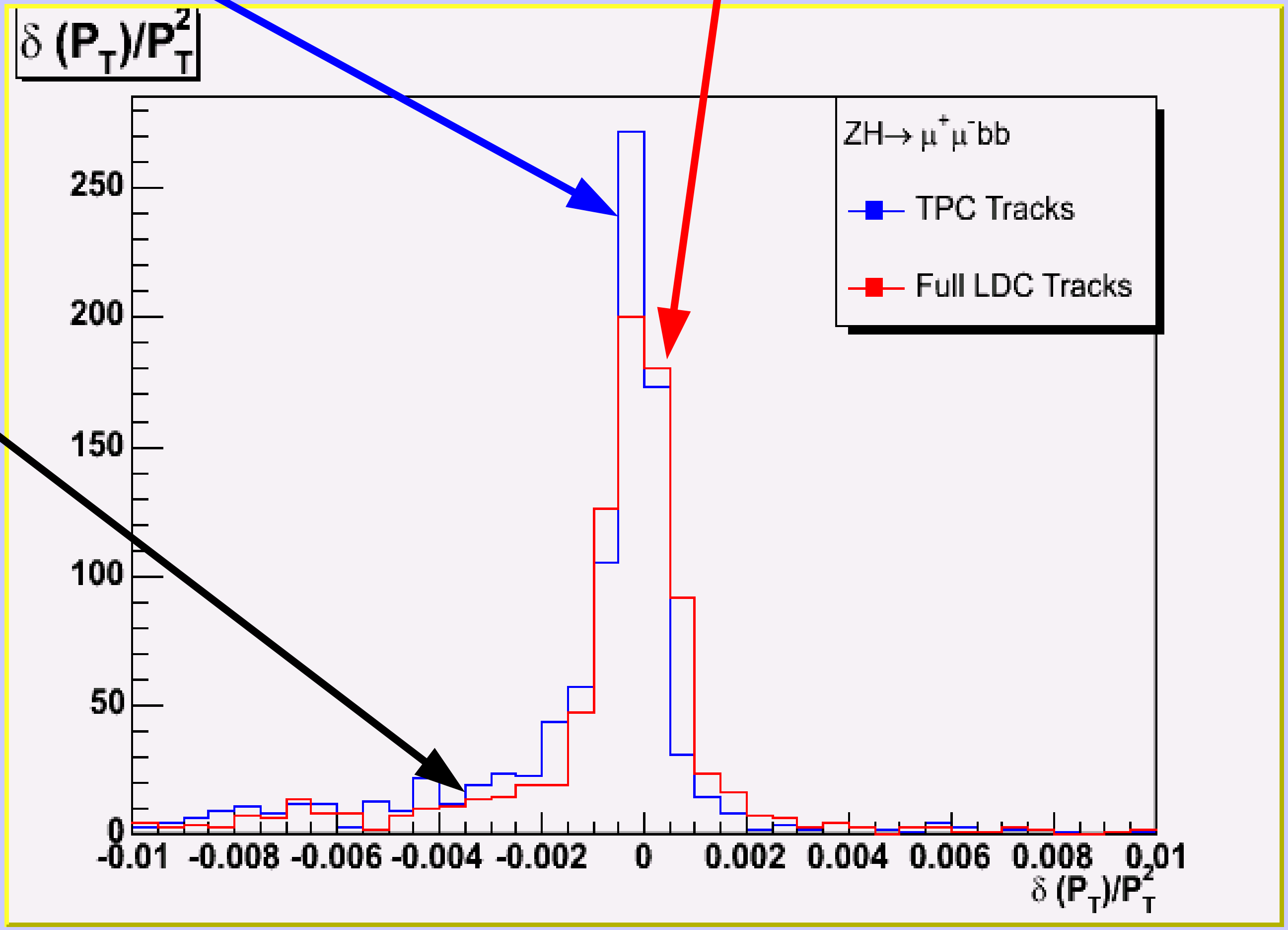
Momentum Resolution

TPC Tracks (Delphi fitting routine) ⇨
$\delta p_T/p_T^2 = 2.3 \cdot 10^{-4}$ (central Gaussian)

LDC Tracks (simple helix fit) ⇨
$\delta p_T/p_T^2 = 3.8 \cdot 10^{-4}$ (central Gaussian)

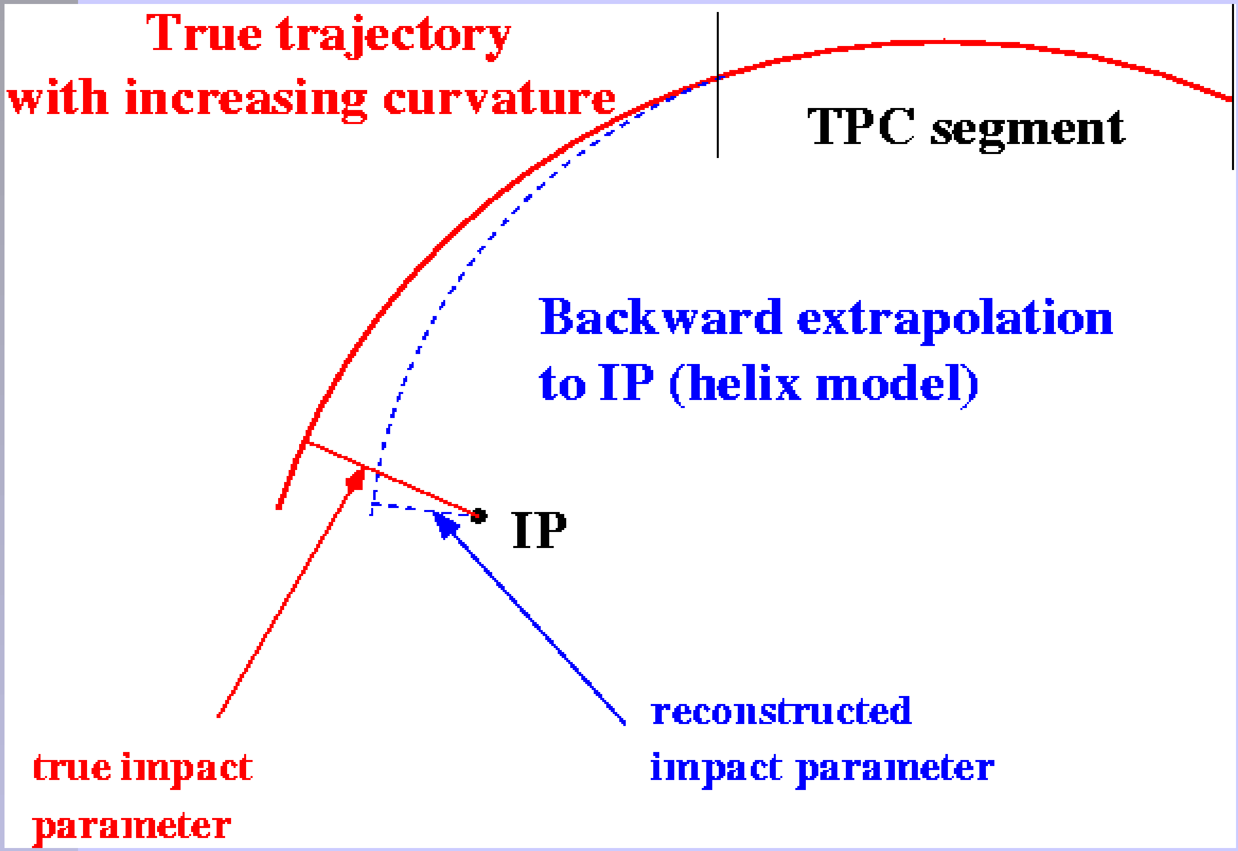
Long tail at left side of distribution ⇨ p_T tends to be underestimated

Explanation on next slide



Momentum resolution of LDC tracks can be improved by using Delphi fitting routine which accounts for energy loss and multiple scatering (work is in progress)

Bias in p_T Reconstruction



- Energy loss ⇒ track does not represent ideal helix, curvature increases along trajectory
- Simple helix fit of full track (VTX+TPC segments) tends gives larger value for curvature compared to true value at starting point of track
- Accurate fit of only TPC track segment also overestimates true track curvature at the PCA
- As a result, in both cases transverse momentum is underestimated

- Energy loss effect also explains bias in reconstruction of impact parameter when the fit is applied only to the TPC track segment

Track Fitting (Simple Helix Fit)

- Brahms tracking package provide a variety of methods to perform track fitting (adopted FORTRAN LEP code).
- Rewriting code in C++ is quite a challenge ⇒ use instead C++ interface to existing FORTRAN implementations (work is ongoing @ MPI to write track fitting code in C++)
- Simple helix fit is performed with routine tfithl.F
- Can be called within C++ code

```
extern "C" {
  void tfithl_(int & NPT, double * XF,double * YF, float * RF, float *PF,
               double * WF,float * ZF,
               float * WZF, int & IOPT, float * VV0,
               float * EE0, float & CH2PH, float & CH2Z);
}
```

- Fits in R-Φ & S-Z projections are disentengled
- No energy loss, MS are taken into account ➞ track parameters are universal / constant along track

Delphi Kalman Fitter

- A more sophisticated fit is provided by Delphi Kalman fitting routine (FORTRAN routine trkfit.F)
- A C++ interface is provided by C++ routine (part of MarlinUtil package)

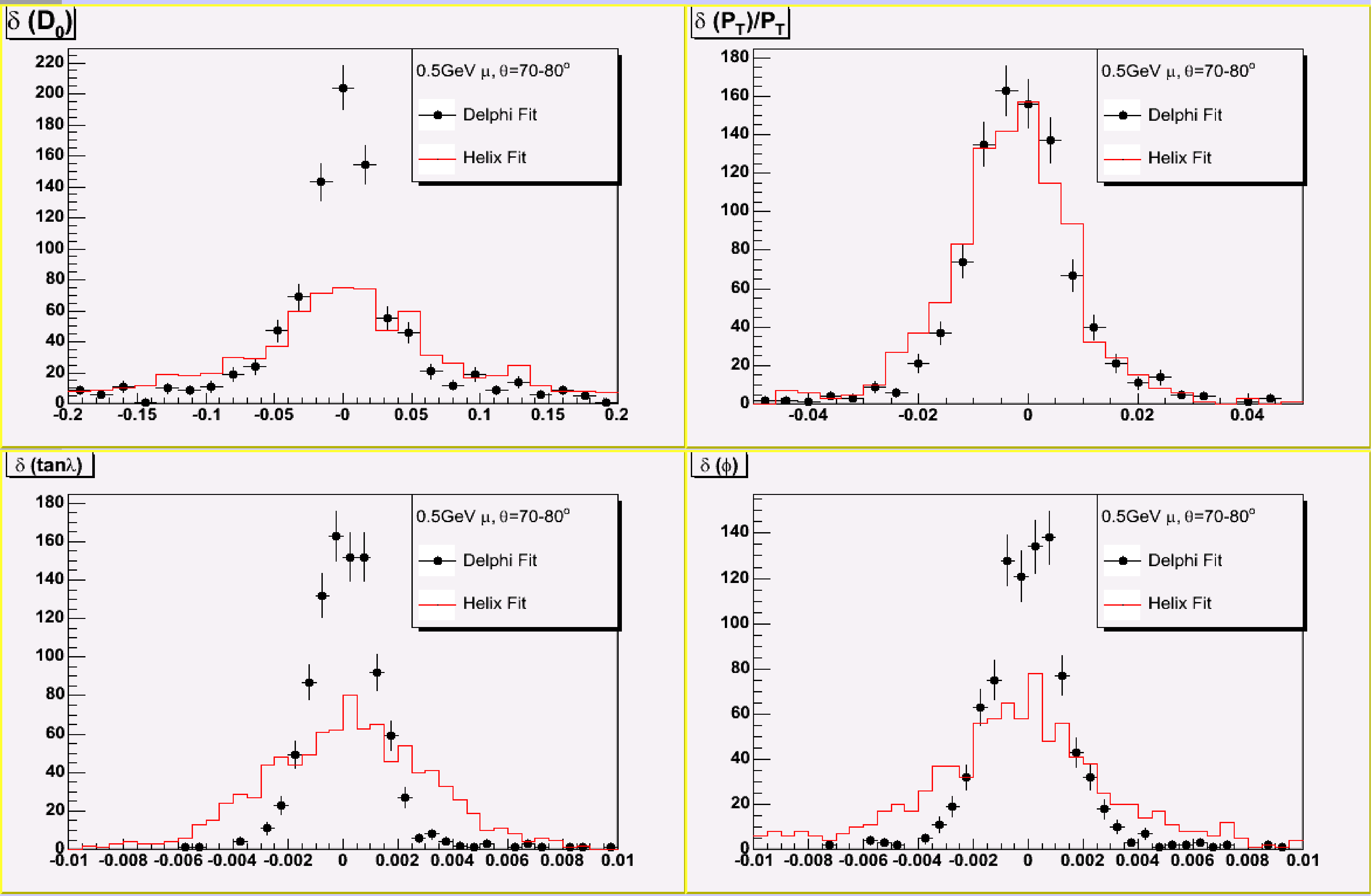
```
int TrackFitting(int & nhits, float bField, int * idet, int * itype,
                 float * x, float * y, float * z, float * RPReso,
                 float * ZReso, float xRef, float yRef, float zRef,
                 int iopt,
                 float * param, float * eparam, float & chi2, int & ndf)
```

- Fit takes into account energy loss and MS ⇨ track parameters vary along track
- Track parameters @ a given point are accessed by specifying reference point (float xRef, float yRef, float zRef)
- Each hit must be assigned detector ID and type (hit in planar detector or in cyllindrical one : int * idet, int * itype)

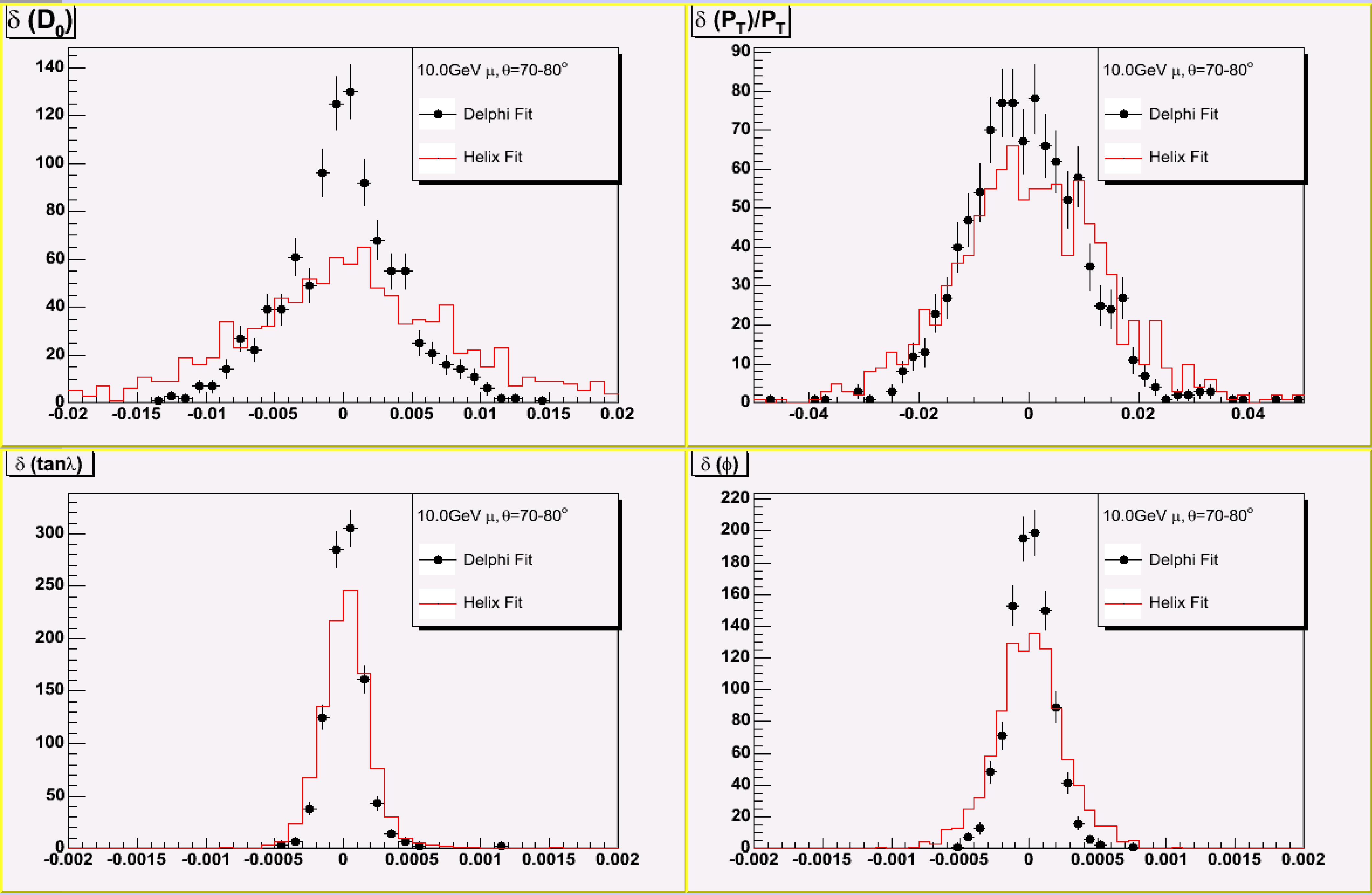
Material Database

- Delphi fitting routine needs info on detector materials and geometry
- In original code these are represented in the form of infinitely thin geometrical shapes (cyllinders & discs orthogonal to beam axis)
- Original Delphi code is extended to handle also rectangular planar detector with arbitrary orientation and position in space and conical materials
- Information on materials is kept in FORTRAN common blocks
- Dedicated Marlin Processor has been developed to build material database / populate common blocks (Marlin Processor “MaterialDB”)
- Processor reads GEAR xml file with description of detector geometry & material properties and transfers this info in FORTAN common blocks

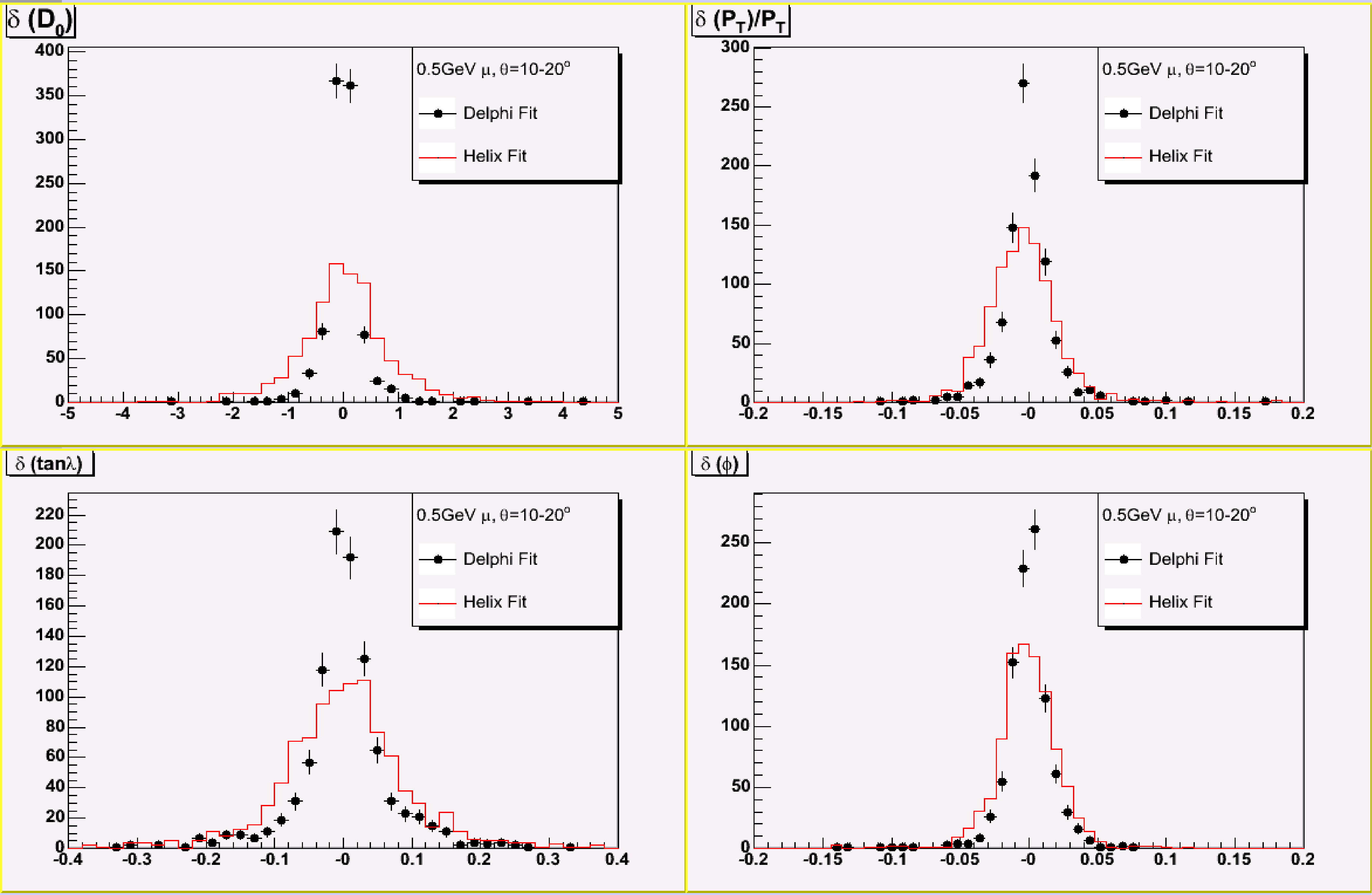
Track Fit Performance (VXD+SIT)



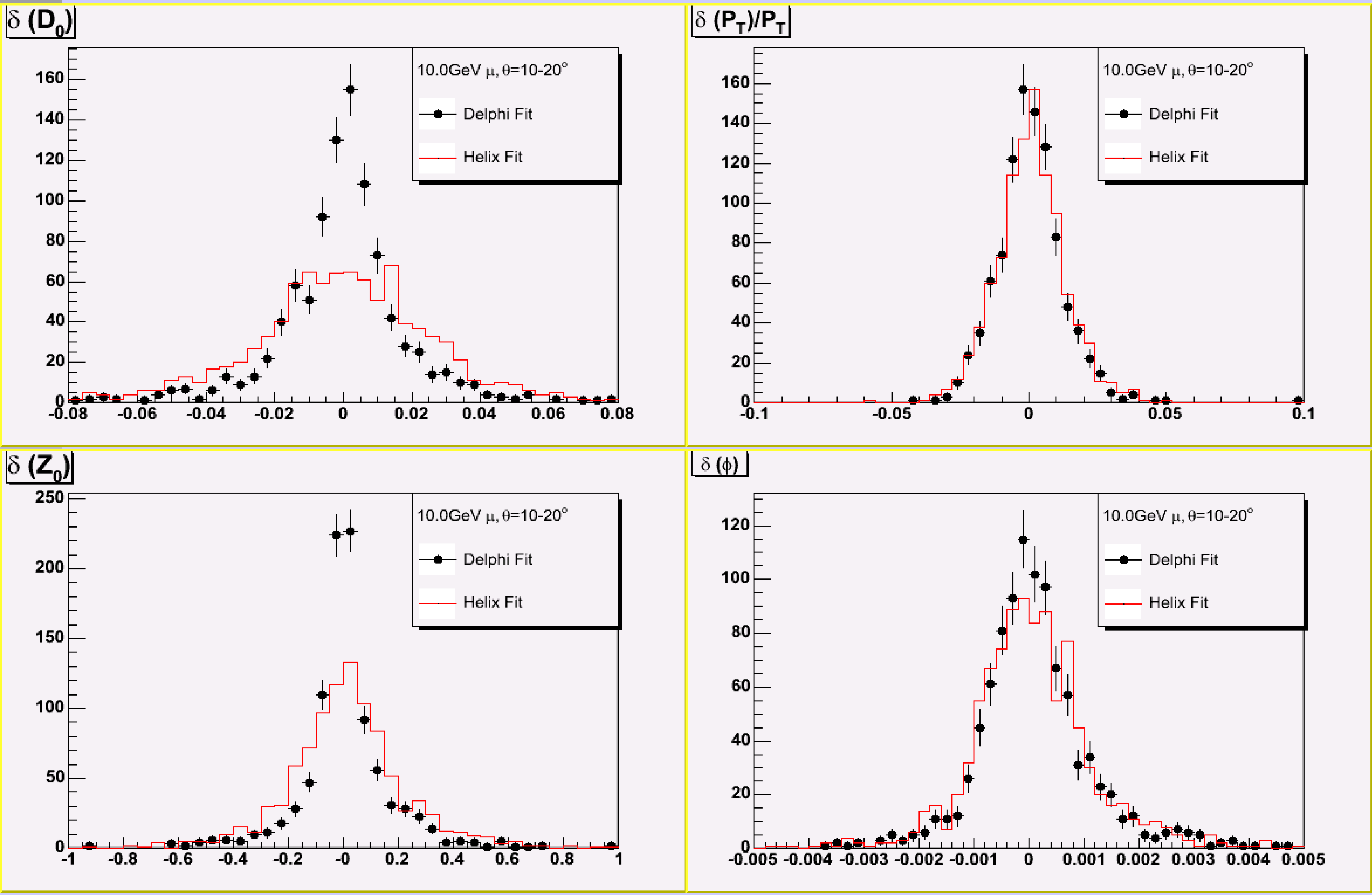
Track Fit Performance (VXD+SIT)



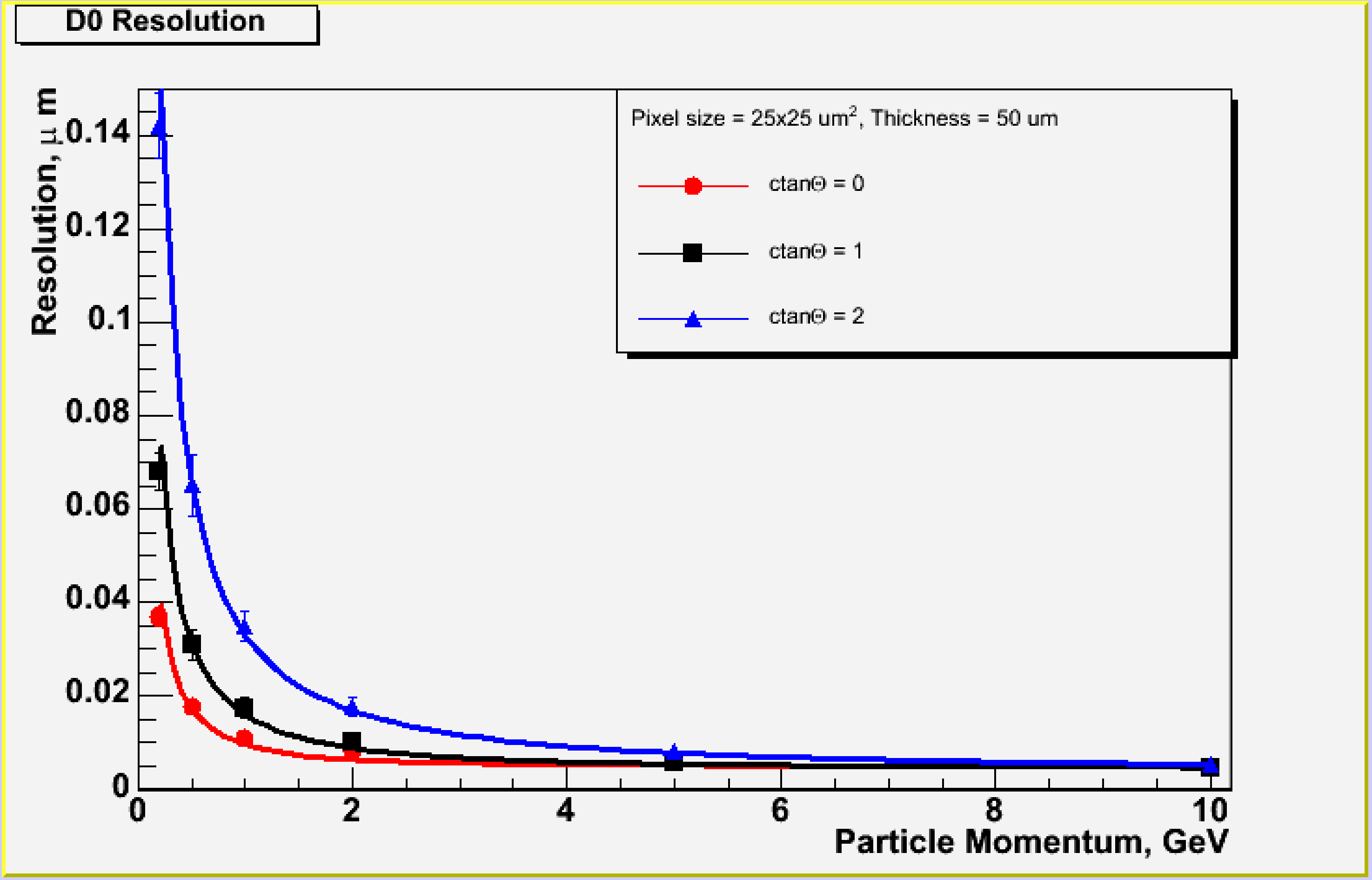
Track Fit Performance (VXD+FTD)



Track Fit Performance (VXD+FTD)



VXT Performance
Impact Parameter resolution



- $\sigma(d_0)=4.6\mu m \oplus 8.6\mu m / p_T[GeV/c] \cdot \sin^{3/2}\theta$
- Fitting of VTX segment of track is done with Delphi routine
- Represantative example : DEPFET VTX
  point resolution 4μm, 5 Layers of 50μm thick sensors

Instruction for those who would like to use tracking software in their studies

- Get the latest versions of the following programs
  - ✔ Marlin (v00-08 or higher), LCIO (v01-07 or higher) and GEAR (v00-03, includes description of VTX geometry)
- In your Marlin root directory create subdirectory packages and checkout MarlinReco and MarlinUtil packages:
  - ✔ `export CVS_RSH=ccvsshcvs;`
  - ✔ `export CVSROOT=:ext:anonymous@cvssrv.ifh.de:/marlinreco`
  - ✔ `cd packages; cvs co MarlinReco; cvs co MarlinUtil;`
- In the directory MarlinReco you find env.sh script. Copy it to marlin root directory, modify as appropriate (scripts sets environment variables indicating paths to LCIO, GEAR *etc*) and source it.
- Change to Marlin root directory and compile the whole code (`gmake`)
- Directory `packages/MarlinReco/examples` contains README file, marlin steering file `marlin_ldctracking.steer`, GEAR xml file `gear_ldc01.xml` and LCIO file ZHllX_350.slcio (Mokka simulation of the LDC detector response to $ZH \rightarrow l^+ l^- X$ events; file contains SimCalorimeterHit and SimTrackerHit collections; Mokka model LDC01).
- Example job (`marlin marlin_ldctracking.steer`) will produce output.slcio file with collections of the reconstructed Si and Full LDC tracks

Conclusions & Outlook

- First implementation of tracking in Si detectors (VTX + FTD + SIT) and full tracking in LDC detector (Zeuthen CVS Repository)
- Full LDC Tracking performance in numbers (first look)
  - Track reconstruction efficiency : 97.3% (for tracks with $p_T>0.5$GeV), 98.9% (for tracks with $p_T>1$GeV)
  - Fake track rate below 0.5%,
  - Splitted track rate 2% (mainly loopers in TPC); needs to be properly handled
  - Momentum resolution :
    - $\delta p_T / p_T^2 \sim 2.3 \cdot 10^{-4}$; TPC tracks, fit accounting for energy loss and multiple scattering)
    - $\delta p_T / p_T^2 \sim 3.8 \cdot 10^{-4}$; Full LDC tracks, still simple helix fit
  - Impact parameter resolution $4.6\mu m \oplus 8.6\mu m / p_T[GeV/c] \cdot \sin^{3/2}\theta$
- Code is still under development. Patrec is in place, but implementation of more accurate fit of full LDC tracks is in progress